CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended) A method of file access control comprising:

[[a.]] storing an encrypted filename of a file at a location in a computing system;

[[b.]] converting the encrypted filename into a plaintext filename;

[[c.]] modifying the plaintext filename into a modified filename; and

[[d.]] authorizing an entity to access the file for performing a write~~type of~~ operation on the file by comparing~~based on~~ the modified filename to the stored encrypted filename.

2. (currently amended) The method according to claim 1, wherein said converting comprises using a key that comprises a combination of two encryption keys to convert the encrypted filename into the plaintext filename.

3. (original) The method according to claim 2, wherein said modifying comprises using a first one of the two encryption keys to encrypt the plaintext filename into the modified filename.

4. (original) The method according to claim 3, wherein said authorizing comprises using the second one of the two encryption keys to encrypt the modified filename to form a result and determining whether the result matches the encrypted filename.

5. (original) The method according to claim 2, wherein said modifying comprises using a first one of the two encryption keys to encrypt the plaintext filename and performing a hash function on the filename thereby forming the modified filename.

6. (original) The method according to claim 5, wherein said authorizing comprises comparing the modified filename to a stored hash value.

7. (original) The method according to claim 1, wherein said encrypted filename is encrypted using a first key prior to said storing and further comprising storing a second encrypted filename of the file at the location wherein the second encrypted filename is encrypted using a second key prior to said storing.

8. (original) The method according to claim 7, wherein said converting comprises using the first key to convert the encrypted filename into the plaintext filename.

9. (original) The method according to claim 8, wherein said modifying comprises using the second key to encrypt the plaintext filename into the modified filename.

10. (original) The method according to claim 9, wherein said authorizing comprises comparing the modified filename to the second encrypted filename.

11. (original) The method according to claim 10, wherein said modifying further comprises performing a hash function on the filename after using the second key to encrypt the plaintext filename.

12. (currently amended) The method according to claim 1, wherein the plaintext filename permits read access to the file~~ and wherein said type of operation is a write operation~~.

13. (original) The method according to claim 1, wherein said storing comprises substituting said encrypted filename into a directory structure at the location in place of the plaintext filename.

14. (original) The method according to claim 1, further comprising encrypting data of the file.

15. (currently amended) An apparatus for controlling access to a file, comprising:

[[a.]] a server for the storing an encrypted filename associated with a file; and

[[b.]] a client in communication with the server for retrieving the encrypted filename from the server, for converting the encrypted filename into a plaintext filename and for modifying the plaintext filename into a modified filename,

wherein the client provides the modified filename to the server and wherein the server determines whether the client is authorized to perform a write~~type of~~ operation on the file by comparing ~~based on~~ the modified filename received from the client to the stored encrypted filename.

16. (currently amended) The apparatus according to claim 15, wherein the plaintext filename permits read access to the file~~ and wherein the type of operation to the file is a write operation~~.

17. (currently amended) The apparatus according to claim 15, wherein said client converts the encrypted filename into the plaintext filename using a key that comprises a combination of two encryption keys.

18. (original) The apparatus according to claim 17, wherein said client forms the modified filename using a first one of the two encryption keys to encrypt the plaintext filename.

19. (currently amended) The apparatus according to claim 18, wherein said server determines whether the client is authorized to perform the write ~~type of~~ operation on the file by using the second one of the two encryption keys to encrypt the modified filename to form a result and determines whether the result matches the encrypted filename provided by the client.

20. (currently amended) The apparatus according to claim 17, wherein said client forms the modified filename using a first one of the two encryption keys to encrypt the plaintext filename and performs a hash function on the filename thereby forming the modified filename.

21. (currently amended) The apparatus according to claim [[20]]17, wherein said server performs a hash function on the filename to form a result and determines whether the client is authorized to perform the read~~type of~~ operation on the file by comparing the result to a stored hash value.

22. (original) The apparatus according to claim 17, wherein said client forms the modified filename using a first one of the two encryption keys to encrypt the plaintext filename and performs a hash function on the filename to form a result and wherein the server determines whether the client is authorized to perform the type of operation on the file by comparing the result to a stored hash value.

23. (original) The apparatus according to claim 15, wherein the encrypted filename is encrypted using a first key and wherein the server stores a second encrypted filename wherein the second encrypted filename is encrypted using a second key.

24. (original) The apparatus according to claim 23, wherein the client converts the encrypted filename into the plaintext filename using the first key and modifies the plaintext filename into the modified filename using the second key.

25. (currently amended) The apparatus according to claim 24, wherein the server determines whether the client is authorized to perform the write ~~a type~~ of operation on the file by comparing the modified filename to the second encrypted filename.

26. (original) The apparatus according to claim 25, wherein the server performs a hash function on the filename after the client uses the second key to modify the filename.

27. (original) The apparatus according to claim 25, wherein the client performs a hash function on the filename after using the second key to modify the filename.

28. (currently amended) An apparatus for controlling access to a file comprising a server having a stored encrypted filename of a file, the server being in communication with a writer and a reader, the writer being a client of the server and having a first key that permits the writer to write to the file and the reader being another client of the server and having a combination key that comprises a combination of the first key and a second key wherein the combination key permits the reader to read the file.

29. (currently amended) The apparatus according to claim 28, wherein the stored encrypted filename is obtained by encrypting a filename of the file using the combination key~~of the first key and the second key~~.

30. (original) The apparatus according to claim 29, wherein the server determines that the writer is authorized to write to the file by receiving from the writer the filename encrypted using the first key, encrypting the received filename again using the second key thereby forming a twice encrypted filename and comparing the twice encrypted filename to the stored encrypted filename.

31. (original) The apparatus according to claim 29, wherein the server determines that the writer is authorized to write to the file by receiving from the writer the filename encrypted using the first key, applying a hash function to the received filename thereby forming a computed hash value and comparing the computed hash value to a stored hash value.

32. (currently amended) An apparatus for controlling access to a file comprising a server having a first stored encrypted filename of the file and a second stored encrypted filename of the file, the server being in communication with a writer and a reader, the writer being a client of the server and having a first key that permits the writer to write to the file and the server determining whether the writer is authorized to write to the file by receiving from the writer the filename encrypted using the second key and comparing the received filename to the second stored encrypted filename and the reader being another client of the server and having a second key that permits the reader to read the file.

33. (original) The apparatus according to claim 32, wherein the reader decrypts the first stored encrypted filename using the first key.

34. (cancelled)

35. (currently amended) The apparatus according to claim [[33]]32, wherein the server performs a hash function on the received filename before comparing the received filename to the second stored encrypted filename.

36. (new) The method according to claim 2, further comprising:

encrypting the plaintext filename using a key that comprises a combination of two encryption keys; and

comparing a result of this encrypting to the stored encrypted filename to determine whether to permit read access to the file.

37. (new) The method according to claim 36, wherein said modifying comprises using a first one of the two encryption keys to encrypt the plaintext filename into the modified filename

38. (new) The apparatus according to claim 15, wherein the client encrypts the plaintext filename and the server compares the encrypted plaintext filename to its stored encrypted filename to determine whether to permit read access to the file.

39. (new) The apparatus according to claim 38, wherein the client encrypts the plaintext filename to form the encrypted plaintext filename using a key that comprises a combination of two encryption keys and the client encrypts the plaintext filename to form the modified filename using a first one of the two encryption keys.